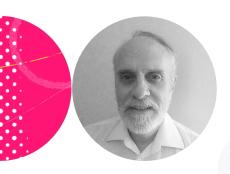
# Course Summary



#### **Gerald Britton**

Pluralsight Author

@GeraldBritton www.linkedin.com/in/geraldbritton



Elements of Reusable Object-Oriented Software

Erich Gamma Richard Helm Ralph Johnson John Vlissides



Cover art © 1994 M.C. Escher / Cordon Art - Baarn - Holland. All rights reserved

Foreword by Grady Booch

Creational

**Structural** 

**Behavioral** 



ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES

#### **SOLID Principles of Object-oriented Design**

Single responsibility

Open/Closed

**Liskov substitution** 

Interface segregation

**Dependency** inversion



#### Don't Repeat Yourself

Don't code it, copy it

Don't copy it, link to it

Use Python modules

Don't link to it, load it

**DRY** 

**Not limited to OOP** 

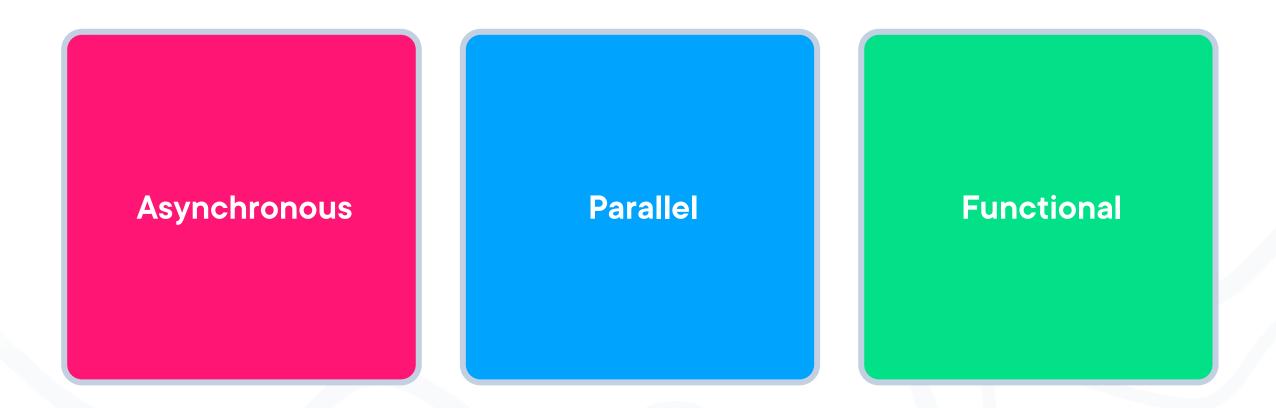
## **Abstract Base Classes**

```
import abc
class AwesomeABC(abc.ABC):
    @abc.abstractmethod
    def must_implement(self, value):
        pass
    def concrete_method(self, value):
        return f'Value is: {value}'
class ThisIsAwesome(AwesomeABC):
    def must_implement(self, value):
        return value * 42
```

awesome\_abc.py



### **Other Design Patterns**



### Thank You!

